<u>**Amendments To the Specification**</u>:

Please replace the paragraph beginning at line 22, page 12 and ending at line 24, page 12, with the following amended paragraph:

Fig. 10c illustrates embodiments of the mux-demux 543a and the multiplexer 543b of the switch of Figs. 10a and 10b. [[ ]]

Figs. 11a and 11b illustrate embodiments of an active SATA to ATA switch in accordance with an embodiment of the present invention.

Please replace the paragraph beginning on line 16, page 13 with the following amended paragraph:

The SATA level 2 host port 210 comprises a PL circuit 211 and a LL circuit 212 and is connected to the outbound high speed differential transmit signals 211tx and to the inbound differential receive signals 211rx and includes a transport receive bus 213r, transport transmit bus 213t, a transport control/status bus 213co generated from the link layer 212 and a control/status bus 213ci being transmitted to the link layer 212. The transport receive bus 213r is connected to the multiplexer 242a. The control/status bus 213co is shown connected to the multiplexer 242b, the transport transmit bus 213t is shown connected to the FIS FIFO output port 245(o1), the OOB signals 211o are [[shownconnected]] <u>shown connected</u> to the switch initialization circuit 244 and to the active host selection circuit 241. The switch initialization circuit 244 generates the Phy ISM control signals 211i.

Please replace the paragraph beginning on line 26, page 13 with the following amended paragraph:

The SATA level 2 host port 220 is shown to comprise a PL circuit 221, and a LL circuit 222, and is connected to the outbound high speed differential transmit signals 221tx and to the inbound differential receive signals 221rx. The port 220 is shown to include a transport receive bus 223r, a transport transmit bus 223t, a transport control/status bus

223co generated from the link layer 222, and a control/status bus 223ci being transmitted to the link layer 222. The transport receive bus 223r is connected to the multiplexer 242a, the control/status bus [[213co]] <u>223co</u> is connected to the multiplexer 242b, the transport transmit bus 223t is connected to a FIS FIFO output port 245(o21), the OOB signals 221o is shown connected to the switch initialization circuit 244 and to the active host selection circuit 241. The switch initialization circuit 244 generates the Phy ISM control signals 221i.

Please replace the paragraph beginning on line 15, page 14 with the following amended paragraph:

The active host selection circuit 241 is the same as the active host selection circuit 141 of Fig. 4. The SFX port selection detection circuits 241a and 241b of Fig. 5 are the same as the port selection detection circuits 141a and 141b, respectively. The active host selection circuit 241generates a multiplexer select signal 241s that selects the input that is placed onto the output of the multiplexer 242a and the multiplexer 242b. The active host selection circuit 241 also generates a host active signal [[141h1]] <u>241h1</u> that when active or at logical state 'one', indicates that the host port 210 is active. The host selection circuit 241 further generates a host active signal [[141h2]] <u>242h2</u> that when active or at logical state 'one', indicates that the host port 220 is active. The host active signals [[141h1]] <u>241h</u> and [[142h2]] <u>242h2</u> serve as input to the demultiplexer 243 and route the control/status bus 233co to the active host.

Please replace the paragraph beginning on line 1, page 15 with the following amended paragraph:

Although the layer 2 switch 200 of Fig. 5 eliminates the timing problems caused by the switch [[200]] <u>100</u> delay, the switch 200 is not able to allow access by two hosts to a single port device via SATA links using standard FIS organization.

Please replace the paragraph beginning on line 17, page 18 with the following amended paragraph:

In the embodiment of Fig. [[5]]6, the arbitration and control circuit 340 intercepts the identify drive response sent from the device and replaces the original value in bits 0 through 4 of word 75 with a new generated value, representing a queue depth value, which is one half of the original queue depth value.

Please replace the paragraph beginning on line 19, page 19 with the following amended paragraph:

- The device sends a Register FIS 40(ii) wherein the REL bit in the Register FIS 40(ii) is set (equals logical 1) and the SERV bit in the Register FIS 40(ii) is reset (equals logical 0) to indicate that the device has queued the command and the command is "released". "Released" is the case wherein the device disconnects ("disconnecting", "connecting", and "reconnecting", are as defined in the Serial ATA specification referred to by reference herein above) after queueing the command for subsequent processing, and reconnects at a [[latter]] later time to complete the command. When the command is released, the host is allowed to send another legacy Read/Write DMA Queued command as long as the queue depth is not exceeded;

Please replace the paragraph beginning on line 3, page 21 with the following amended paragraph:

In the case of legacy queue commands, when the switch 300 receives from the device a Set Device Bits FIS 40(v) wherein the SERV bit in the Set Device Bits FIS 40(v) is set or Register FIS 40(ii) wherein the SERV bit in the Register FIS 40(ii) is set, the switch 300 can not forward the Set Device Bits FIS 40(v) since the tag value of the command that the device needs to service is not yet available. In order to obtain the tag value, the switch 300 sends a SERVICE command to the device, the device responds with a Register FIS 40(ii) including the tag. The switch 300 then remaps the tag to identify the host and the original host tag value. If there is no legacy queue command pending from the host in the switch 300, the switch then sends Set Device Bits FIS 40(v) to the host with the SERV bit set. If there is a legacy queue command pending in the switch, the switch 300 stores the command in [[internal storage]] pending task file 344 and when the device is

ready to release the pending legacy queue command, the device responds with a Register FIS 40(ii) wherein the REL bit and the SERV bit in the Register FIS 40(ii) are set. When the host responds with a SERVICE command, the switch 300 responds with a Register FIS 40(ii) including the original host tag value.

Please replace the paragraph beginning on line 17, page 21 with the following amended paragraph:

In the case of native queue commands, when the switch 300 receives a DMA Setup FIS 40(vi) from the device, the switch 300 first remaps the tag in the DMA Setup [[FIS 40(vi)(ii)]] <u>FIS 40(iv)(ii)</u> to identify the host and the original host tag value and then forwards the DMA Setup [[FIS 40(vi)(ii)]] <u>FIS 40(iv)(ii)</u>, with the tag replaced with the original host tag, to the identified host.

Please replace the paragraph beginning on line 1, page 22 with the following amended paragraph:

In idle state 361, if a legacy queue command is received from either hosts or there is a pending legacy queue command in the Pending Task File <u>344</u> [[362]], then the switch 300 changes state to the host-selection state 363. Otherwise, if the Set Device Bits FIS 40(v) or the Register FIS 40(ii) with the SERV bit set is received from the device 372, then the switch 300 [[ ]]changes state to [[send-the-service state]] <u>send service CMD state</u> 373. Otherwise, the switch [[ ]]remains in idle state 361.

Please replace the paragraph beginning on line 10, page 22 with the following amended paragraph:

In the send-LQ CMD state 364, the switch 300 first maps the selected LQ host tag to a send device tag, replaces the selected LQ host tag with the send device tag and then forwards the pending command to the device. The switch 300 then changes state to [[wait for the device response state]] <u>wait for device response state</u> 365. The send device tag refers to a tag that is sent to the device by the switch 300.

Please replace the paragraph beginning on line 15, page 22 with the following amended paragraph:

In the [[wait-for-the-device-response state]] <u>wait for device response state</u> 365, if the device response is received, then the switch 300 changes state to the check-device-response state 366. Otherwise, the switch 300 remains in the [[wait-for-the-device-response state]] <u>wait for device response state</u> 365. In the check-device-response state 366, if the device response is Register FIS 40(ii) with the REL bit set and the SERV bit set in the Register FIS 40(ii), the switch 300 changes state to the disconnect/reconnect state 366b. Otherwise, if the device response is Register FIS 40(ii) with REL bit set and SERV bit reset, the switch 300 changes state to disconnect state 366d. Still Otherwise, if the device response is Data FIS 40(vii) or DMA Activate FIS 40(iii), the switch 300 changes state to the execute state 366f. Yet otherwise, if the device response is Register FIS 40(ii) with ERR bit set, the switch 300 changes state to the error state 366h. Still otherwise, the switch changes state to the discard state 366j and discards the received FIS and changes state to the idle State 361.

Please replace the paragraph beginning on line 5, page 23 with the following amended paragraph:

In the send-service CMD state 373, the switch 300 sends the service CMD to the device and changes state to the [[wait-for-the-device-tag state]] <u>wait for device tag state</u> 374.

Please replace the paragraph beginning on line 7, page 23 with the following amended paragraph:

While in the [[wait-for-the-device-tag state]] <u>wait for device tag state</u> 374, if the device response is received, the switch 300 changes state to the remap state 375, otherwise, the switch 300 remains in the [[wait-for-the-device-tag state]] <u>wait for device tag state</u> 374. In the remap state 375, the switch 300 remaps a receive<u>d</u> device tag, the tag that is received from the device by the switch 300, to identify the host and the original host tag and additionally, the switch 300 replaces the receive<u>d</u> device tag with the original host

tag. If there is a pending queue CMD in the identified host task file, the switch 300 changes state to the save state 376b, otherwise, the switch 300 changes state to the [[reconnect-to-the-host state]] <u>reconnect to host state</u> 376a. In the [[reconnect-to-the-host state 376b]] <u>reconnect to host state</u> <u>376a,</u>, the switch 300 sends a Set Device Bits FIS 40(v) with the SERV bit set to an identified host (a host that was identified in the remap state 375) and then changes state to the [[wait-for-the-host-response state]] <u>wait for the host response state</u> 377.

Please replace the paragraph beginning on line 18, page 23 with the following amended paragraph:

In the [[wait-for-the-host-response state]] <u>wait for host response state</u> 377, if the identified host's response is received, then the switch 300 changes state to the check-host-response state 378, otherwise, the switch 300 remains in the [[wait-for-the-host-response state]] <u>wait for host response state</u> 377. In the check-host-response state 378, if the host response is a service command, the switch 300 changes state to the send-tag-to-host state 378b, otherwise, if the host response is another LQ CMD, then the switch 300 changes state to the set-pending state 378c, otherwise, the switch 300 changes state to the error 2 state 378d.

Please replace the paragraph beginning on line 25, page 23 with the following amended paragraph:

In the [[send-tag-to-host state]] <u>send tag to host state</u> 378b, the switch 300 sends a Register FIS, with the original host tag, to the identified host and changes state to the reconnected state 379. In the [[set-pending state]] <u>set pending state</u> 378c, the switch 300 sets a pending queue CMD flag to indicate that the host has sent another legacy queue command. In the save state 376b, the switch 300 saves the task file of the identified host in the pending task file and then sends a Register FIS 40(v), with the SERV bit set and the REL bit set to the identified host.

Please replace the paragraph beginning on line 31, page 23 with the following amended paragraph:

In the disconnect/reconnect state 366b, the reconnect to host state 376a, the [[save-state]] <u>save state</u> 376b, and the [[send-tag-to-host state]] <u>send tag to host state</u> 378b, a modified FIS or a new FIS aresent to the host. The arbitration and control circuit 340 generates the task file corresponding to the modified FIS (or the new FIS) and transmits the same onto the control task file output bus 354i, which is connected to an input of the mux-demux 354. The circuit 340 further sets the value of the control signal 354c to select the bus 354i and demultiplexes the same to serve as the output of the mux-demux 354.

Please replace the paragraph beginning on line 31, page 23 with the following amended paragraph:

In the disconnect/reconnect state 366b, the reconnect to host state 376a, the save-state 376b, and the send-tag-to-host state 378b, a modified FIS or a new FIS [[aresent]] <u>are sent</u> to the host. The arbitration and control circuit 340 generates the task file corresponding to the modified FIS (or the new FIS) and transmits the same onto the control task file output bus 354i, which is connected to an input of the mux-demux 354. The circuit 340 further sets the value of the control signal 354c to select the bus 354i and demultiplexes the same to serve as the output of the mux-demux 354.

Please replace the paragraph beginning on line 6, page 24 with the following amended paragraph:

In the [[send-LQ CMD state]] <u>send LQ CMD state</u> 364, the send service CMD 373, the modified FIS (or the new FIS) [[aresent]] <u>are sent</u> to the device. The arbitration and control circuit 340 generates the task file corresponding either to the modified FIS or to the new FIS on the device control task file output bus 352i, which in turn is connected to one of the inputs of the multiplexer 352, as shown in Fig. 5, and sets the value of the

select signal 352s to select the bus 352i as the output of the multiplexer 352, which in turn is connected to the device task file input bus 336i.

Please replace paragraph beginning on line 13, page 24 with the following amended paragraph:

Fig. 8b shows a flow chart of the operation of the switch 300 for the native queue commands (NQ CMDs). In the idle state 381, a number of decisions may be made as shown at 382-386. At 382, if a native queue command is received from either hosts and the device has responded to the previous NQ CMDs, then switch 300 changes state to the host-selection state 382a, otherwise, at 383, if the Register FIS, with the ERR bit reset, is received from the device, then the switch 300 changes state to the NQ-disconnect state 383a. Otherwise, at 384, if the DMA Setup FIS 40(iv) is received from the device, then the switch 300 changes state to the [[NQ-remap state]] <u>NQ remap state</u> 384a. Still otherwise, at 385, if [[ ]]Set Device Bits FIS 40(v) is received, and at 385a, the ERR bit in the Set Device Bits FIS is reset, then the switch changes state to the [[NQ-status state]] <u>NQ status state</u> 385b, otherwise, if the ERR bit is set, then the switch 300 changes state to the [[NQ-error state]] <u>NQ error state</u> 386a. If at 385, the Set Device Bits FIS does not indicate completion status, at 386, if a Device Register FIS, with the ERR bit set, is received, then the switch 300 changes state to the NQ-error state 386a, otherwise, the switch 300 remains in the idle state 381.

Please replace the paragraph beginning on line 27, page 24 with the following amended paragraph:

In the [[host-selection-state]] <u>host selection state</u> 382a, the switch 300 arbitrates between the hosts and selects the host whose pending command will be subsequently forwarded to the device in the send-NQ CMD state 382b. The switch 300 then changes state to the send NQ CMD state 382b.

Please replace the paragraph beginning on line 31, page 24 with the following amended paragraph:

In the [[send-NQ CMD state]] send NQ CMD state 382b, the switch 300 first maps the selected NQ host tag to a send device tag, replaces the selected NQ host tag with the send device tag, forwards the command that was sent in send-NQ CMD state 382b to the device and sets a flag, "device _not_responded", and changes state to the idle state 381. The flag, "device_not_responded" , indicates that the device has not yet responded to a native queue command.

Please replace the paragraph beginning on line 4, page 25 with the following amended paragraph:

In the [[NQ-disconnect state]] NQ disconnect state 383a, the switch 300 causes the Register FIS to be forwarded to the selected host, resets the flag "device_not_responded", and then changes state to the idle state 381. In the [[NQ-remap state]] NQ remap state 384a, the switch 300 remaps the receive device tag to identify the host and the original host tag, and replaces the receive device tag with the original host tag in the DMA Setup FIS and sends the DMA Setup FIS to the identified host, and changes state to the [[NQ-reconnected state]] NQ reconnected state 384b. In the [[NQ-reconnected state]] NQ reconnected state 384b, the identified host is reconnected to the device, and the Data FIS is transferred between the reconnected host and the device. In the [[NQ-reconnected state]] NQ reconnected state 384b, the switch 300 checks as to whether or not the DMA transfer count is exhausted at 384c. If the DMA transfer count is not exhausted, the switch 300 remains in the reconnected state 384b, otherwise, the switch 300 changes state [[ ]]to idle state 381. In the [[NQ-status state]] NQ status state 385b, the switch 300 processes the status of successfully completed NQ CMDs, which is reported by the device in the Sactive field 41 of the Set Device Bits FIS 40(v).

Please replace the paragraph beginning on line 17, page 25 with the following amended paragraph:

The switch 300 generates a host 11 Sactive field and a host 12 Sactive field from the Sactive field 41 such that the [[ ]]host 11 Sactive field includes only the tags in the

Sactive field 41 that belong to the host 11 and the host 12 Sactive field includes only the tags in the Sactive field 41 that belong to the host 12. The switch 300 forwards the Set Device Bits FIS to the host 11 with the Sactive field 41 replaced with the host 11 Sactive field, and then forwards the Set Device Bits FIS to the host 12 with the Sactive field 41 replaced with the host 12 Sactive field. The switch 300 then changes to the idle state 381. In the [[NQ-error state]] NQ error state 386, the switch 300 performs error handling and after completion of error handling, changes state to the idle state 381.

Please replace the paragraph beginning on line 26, page 25 with the following amended paragraph:

In the NQ-remap state 384a and the [[NQ-status state]] NQ status state 385b, a modified FIS is sent to the host. The arbitration and control circuit 340 generates the task file corresponding to the modified FIS or the new FIS and transmits on control task file output bus [[354]] 354i , that is connected to the second input of mux-demux 354 and sets the value on the control signal 354c to select and demultiplex the task file output bus 354i to the selected host.

Please replace the paragraph beginning on line 31, page 25 with the following amended paragraph:

In the [[send-NQ CMD state]] send NQ CMD state 364, a modified FIS is sent to the device. The arbitration and control circuit 340 generates the task file corresponding to the modified FIS and transmits on device control task file output bus 352i that is connected to one of the inputs of the multiplexer 352 and sets the value on the select signal 352s to select the bus 352i as the output of multiplexer 352, which is connected to the device task file input bus 336i.

Please replace the paragraph beginning on line 22, page 26 with the following amended paragraph:

Referring to Fig. 7a, the arbitration and control circuit 340 comprises a host arbitration circuit 343, Tag/Sactive mapping circuit 341, and control circuit 342. [[Thefunctions]] <u>The functions</u> performed by the Tag/Sactive Mapping circuit 341 include:

Please replace the paragraph beginning on line 30, page 27 with the following amended paragraph:

- generating control signal [[342c]] <u>343c</u> for host arbitration circuit 343;

Please replace the paragraph beginning on line 3, page 28 with the following amended paragraph:

Fig. 7b [[ ]]shows the Tag/Sactive mapping circuit <u>341</u>, used in one of the [[embodiment]] <u>embodiments</u> of the present invention. The Tag/Sactive Mapping circuit 341 includes a tag memory 341d, a valid LQT register 341a for indicating whether or not the corresponding LQT is valid, a LQT map 341b, a NQT map 341g, a NQT inverse map 341f, a device tag multiplexer 341m1, [[and]] a host tag multiplexer 341m2<u>, a retrieve tag register 341e, and a Sactive Map 341s</u> . The Tag/Sactive mapping circuit 341 inputs includes device tag input 341j, a host tag input 341i, a Sactive input 341k, host queue depth input 341qd, and a control bus 341ctl. The Tag/Sactive mapping circuit 341 generates certain outputs including a mapped host tag 341dt, a retrieved host tag 341ht, a host 11 Sactive output bus 341s1 and a host 12 Sactive output bus 341s2.

Please replace the paragraph beginning on line 17, page 28 with the following amended paragraph:

This operation of mapping the NQ tag is performed by the NQT map 341g. The NQT map 341g receives a selected host tag input <u>341i</u> and a host queue depth 341qd, and its output <u>341nt</u> is connected to an input of the device tag multiplexer 341m1. If the selected host tag input 341i is from the host 11 (signal 341h is logical zero), then the output of the NQT map is equal to the selected host tag input 341i, otherwise if the host

tag input 341i is from the host 12 (signal 341h is logical one), then the output of the NQT map is equal to the [[ ]]selected host tag input 341i plus the host queue depth.

Please replace the paragraph beginning on line 3, page 29 with the following amended paragraph:

In the case of legacy queue commands, the host tag values are between 0 and 31 regardless of the host queue depth. As mentioned above, the device tags from 0 to host_queue_depth_minus_one are assigned to the host 11 range, and device tags from host_queue_depth to (2* host_queue_depth -1) are assigned to the host 12 range. A tag memory unit 341d is used to store the host tag values corresponding to the device tags. This reduces the complexity associated with the function performed by the reverse mapping in that the tag memory unit 341d is accessed at the address corresponding to the received device tag.

Please replace the paragraph beginning on line 10, page 31 with the following amended paragraph:

- [[H1_Qempty]] H2_Qempty when set indicates host [[11]]12 has an empty queue, when reset a non-empty queue

Please replace the table beginning on line 12, page 31 with the following amended table:

Table 1. Host Arbitration Operation

	H1_fis_req	H2_fis_req	H1_Qcmd	H2_Qcmd	H1_Qempty	H2_Qempty	Host Arbitration Action
1	1	0	x	x	1	1	Grant to host 11
2	0	1	x	x	1	1	Grant to host 12
3	1	1	0	0	1	1	Grant to host with the priority,
4	1	1	0	1	1	1	Grant to host 11
5	1	1	1	0	1	1	Grant to host 12
6	1	1	1	1	1	1	Grant to host with the priority,
7	1	0	x	x	0	1	Grant to host 11

8	0	1	x	0	0	1	Grant is not issued [3]
9	0	1	x	1	0	1	Grant to host 12
10	1	1	0	0	0	1	Grant to host 11[1]
11	1	1	0	1	0	1	Grant to host 11 [1]
12	1	1	1	0	0	1	Grant to host 11. Alternatively if legacy [[cueue]] queue command then Grant to host 11, elseif native queue command no Grant is issued [4]
13	1	1	1	1	0	1	Grant to host with the priority
14	1	0	0	x	1	0	Grant is not issued [3]
15	1	0	1	x	1	0	Grant to host 11
16	0	1	x	x	1	0	Grant to host 12
17	1	1	0	0	1	0	Grant to host 12 [2]
18	1	1	0	1	1	0	Grant to host 12. Alternatively if legacy queue command Grant to host 12, elseif native queue command no Grant is issued [4]
19	1	1	1	0	1	0	Grant to host 12 [2]
20	1	1	1	1	1	0	Grant to host with the priority
21	1	0	0	x	0	0	Grant to host 11 [1]
22	1	0	1	x	0	0	Grant to host 11
23	0	1	x	0	0	0	Grant to host 12 [2]
24	0	1	x	1	0	0	Grant to host 12
25	1	1	0	0	0	0	Grant to host with the priority
26	1	1	0	1	0	0	Grant to host 12
27	1	1	1	0	0	0	Grant to host 11
28	1	1	1	1	0	0	Grant to host with the priority
29	0	0	x	x	x	x	Grant is not issued.

Notes:

(1) Host 11 issues a non-queue command while it has a non-empty queue. The Switch will forward the command to Device. In response to receipt of non-queue command with non-empty queue the Device will set Error (ERR). Receipt of Error with non-empty queue will cause the Switch to flush non-empty queue commands and sending ERR status to Hosts with non-empty queue.

(2) Host [[11]] 12 issues a non-queue command while it has a non-empty queue. The Switch will forward the command to Device. In response to receipt of non-queue command with non-empty queue the Device will set Error (ERR). Receipt of Error with non-empty queue

will cause the Switch to flush non-empty queue commands and sending ERR status to Hosts with non-empty queue.

(3) Since the Host sending the non-queue command has an empty queue and the other Host has a non-empty queue sending the non-queue command will cause the Device to set Error and result in queue being Flushed [[therefore]] . Therefore when the sending [[the]] Host has and empty queue and sends a non-queue command while the other Host has a non-empty queue, the command is held until the other Host queue is emptied.

(4) As mentioned earlier when a Host with an empty queue issues a non-queue command while the other Host has a non-empty queue, the non-queue command is held until the queue is emptied. In this case in order to allow the queue to empty when the Host with non-empty queue sends another queue command it is desirable to hold the newly received queue command until the queue is emptied and the non-queue command is sent. In the case of a Legacy Queue Command it is not practical to hold the newly received legacy queue command, since the Switch has to release it when the Device is reconnecting. However this limitation does not apply to native queue command, and in case of native queue command

Please replace the paragraph beginning on line 19, page 35 with the following amended paragraph:

Figs. 10a and 10b show block diagrams of another embodiment of the active switch 500 of present invention. One of the features of the architecture of the active switch 500 is use of a common FIFO 555a, 555b for passing payload of Data FIS among the SATA ports without passing Data through the layer 4, thus reducing the delay associated with the switch as well as the number of FIFOs. Another feature of the active switch 500 is the FIS bus structure that allows passing non-Data FIS and first Dword of Data FIS among SATA ports at layer 3 without passing FIS through the layer 4, thereby reducing the delay thru the active switch 500.

Please replace the paragraph beginning on line 10, page 39 with the following amended paragraph:

- mapping a selected host queue tag to a send device tag and in the case of a legacy queue tag saving the result in a tag [[memeory]] memory 341d, and keeping a list of valid queue tags

Please replace the paragraph beginning on line 21, page 41 with the following amended paragraph:

The SATA layer 3 to ATA Bridge 730 comprises [[ ]]a[[ATA]] SATA Transport Layer 733, and a ATA Interface Bridge 732. The ATA Interface [[Bridges]] Bridge 732 is connected to the ATA link 736 and converts (bridges) the activity on the ATA bus 736 to the activity on the Transport layer interface 733io and visa versa. The Transport Layer 733 is the same as the Transport Layer 413 of Fig. 9.[[.]]

Please replace the paragraph beginning on line 28, page 43 with the following amended paragraph:

As shown in Fig. 4, in one of the systems of the prior art, there is a physical layer for one host, another physical layer for the other host and a physical layer for the device or storage unit used by a switch that is coupled between the hosts and the device. None of the other layers are in communication with the hosts and/or device. Through the physical layer, one of the hosts is selected by a multiplexer for communicating [[withthe]] with the device and then the device sends data to that active host. An active host selection circuit decides or selects which host is initially selected along with an initialization circuit. Thus, this prior art switch only needs layer one or the physical layer to communicate, no other layers are needed for communications. However, as noted earlier, one of the problems with such a prior art system is the delay through the switch. Another problem is that only one host can communicate with the device at any given time.

Please replace the paragraph beginning on line 11, page 46 with the following amended paragraph:

In Fig. 11, layers 1 and 2 are replaced with a non-serial ATA port such as an ATA port thereby enabling use of storage units using [[non-seriall]] non-serial ATA standard improving system cost using lower cost storage units in the system.

Please replace the paragraph beginning on line 28, page 46 with the following amended paragraph:

The capitalization of certain letters of names of signals, states, devices and so forth, as used throughout this patent document, are done so to maintain consistency with names of corresponding signals, states, devices and so forth disclosed in the "Serial ATA: High Speed Serialized At Attachment", published by Serial ATA work group www.serialata.com, the contents of which [[areincorporated]] are incorporated herein by reference as though set forth in full.